Int J Adv Manuf Technol (2012) 62:135–145
DOI 10.1007/s00170-011-3809-1

ORIGINAL ARTICLE

On flowshop scheduling problems with the aging effect and resource allocation

Agnieszka Rudek · Radosław Rudek

Received: 15 July 2011 / Accepted: 22 November 2011 / Published online: 21 December 2011

Abstract In this paper, we focus on real-life settings that require the development of new models of flowshop scheduling problems, where job processing times can increase with the number of processed jobs due to the aging effect and decrease by the allocation of additional resource. We analyse the makespan minimization flowshop problem with such model and also with the aging effect only. We prove that the considered problems and their special cases are still polynomially solvable under given conditions, and on their basis, we provide optimal polynomial time solution algorithms.

Keywords Scheduling · Flowshop · Aging effect · Deteriorating · Resource allocation

1 Introduction

In many industrial systems, the processing times of jobs can increase due to deterioration of machines that has a negative influence on the total output. On the other hand, additional resources allocated to jobs can decrease their processing times, but it also increases the production cost (e.g. financial, environmental etc.).

In particular, such problems occur during the pickling process that accompanies galvanization and plays a significant role in various industry sectors, where galvanized steel is an essential fabrication component (e.g. aircraft and car industries). It follows from the fact that metal items have to be cleaned before exposing them to further treatment (see [24]). This cleaning process, called pickling, consists of at least two stages. In the first one, a human worker prepares items and operates the bath, and in the second, the items are exposed to the pickling process that is one of the most important stages in metal finishing [8, 24]. However, the number of stages can be greater, since pickling can consist of two stages on its own [24] or it is preceded by such activities as removal of grease, oil, polishing or drawing compounds etc. [9].

The mentioned method of cleaning metal surfaces is based on immersion of metal items into pickling baths that remove a metal dirt (e.g. metal oxides, heat tread scale and foreign metals) from pickled parts. The composition of a pickling bath (e.g. hydroxides: potassium, sodium; acids: nitric, hydrochloric, hydrofluoric, sulphuric) and its parameters (e.g. metal pickling/cleaning time) depends on a type of a cleaned surface (e.g. steel, copper, aluminium, magnesium alloys); for more details, see [8]. However, during pickling process, the parameters of the bath (an active substance) change. For example, pickling of a stainless steel causes that the concentration of hydrochloric or sulphuric acid in the bath decreases and the concentration of ferrous salts increases—that slows down the etching rate. The measurable result is the time required to clean an individual metal item increases as more dirt (e.g. foreign metals) is dissolved in the pickling bath [19].

When the pickling/cleaning time is too long, the active substance can be added or the bath is changed. However, the active substance, even if it is worn

A. Rudek
Department of Systems and Computer Networks,
Wrocław University of Technology,
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: agnieszka.wielgus@pwr.wroc.pl

R. Rudek (✉)
Wrocław University of Economics, Komandorska 118/120,
53-345 Wrocław, Poland
e-mail: radoslaw.rudek@ue.wroc.pl

out, must be chemically neutralized, since the remains of acid or alkalis can be still active and dangerous for human and the natural environment—especially for ground and surface water [8]. In some cases, the efficiency of pickling can be improved for individual items without adding the active substance or changing the pickling bath. One of such methods is heating up the bath, since its activity can be controlled by temperature. For instance, in the case of a pickling bath based on sulphuric acid, heating it up by 10°C increases the pickling (cleaning) speed by about 70%. To increase the activity of this type of pickle to a maximum, it is heated up to 95°C (see [24]).

At first, note that the pickling process consists of at least two stages; therefore, we can express it as a flowshop scheduling problem. However, the classical flowshop scheduling problems (with constant job processing times) are perceived to be more interesting in a theoretical context than as a practical research [10]. It follows from the observations that algorithms constructed on the basis of the classical models usually provide unsatisfactory (unstable) solutions for real-life flowshop problems, since these models do not take into consideration additional factors that are significant in practice, e.g. sequence-dependent setup times [16, 20] or variable processing times [2, 4, 5, 23, 27, 31]. To this group belong the described real-life problems, where a deterioration of machines (understood as chemical cleaning baths, lathe machines, human workers etc.) or allocation of additional resources to jobs (e.g. a gas that heats up the pickling bath) affects the production parameters such as job processing times. Therefore, to construct efficient solution algorithms, we have to develop more precise models that take into consideration the fundamental aspects of the described systems, i.e. the decreasing efficiency of machines (e.g. the pickling bath) and allocation of additional resources (e.g. a gas that heats up the pickling bath).

In the scheduling theory, there are two approaches to model the first aspect. One assumes that the job processing times are non-decreasing functions of their starting times. In this case, the phenomenon is known as "deteriorating effect" and has been extensively studied in the last decade (e.g. [4]). However, scheduling models consistent with this approach are not relevant to many real-life industrial problems. It is especially significant for flowshop environments, where deterioration does not take place (or it is negligible) during idle times of machines, e.g. the efficiency of the pickling bath does not change or it is negligible if no elements are in the bath. Such inconveniences are absent in the approach called "aging effect" (see [12, 18, 32]) and sometimes "deteriorating" (see [17, 28, 29]), in which job processing times are described by non-decreasing functions dependent on the actual condition (fatigue) of machines affected by already processed jobs. Therefore, in this paper, we will use the aging models to express the decreasing efficiency of a pickling bath. Note that the similarity of jobs usually allows to assume that a machine (e.g. understood as a bath) deteriorates with the number of processed jobs [15, 33].

The second aspect, the improvement of machine efficiency (e.g. of the bath by heating up the pickle using an additional energy), can be modelled by additional resources (e.g. [2, 11, 22]), i.e. gas or electric energy (to heat up the pickle) and/or the active substance. However, the additional usage of the active substances should be avoided, since they are hazardous to the natural environment. Therefore, we will exclude this approach from further consideration.

Therefore, in this paper, we will express the discussed problems as flowshop scheduling problems with the aging effect and additional resource allocation, where the processing time of each job can be described by a function that is non-decreasing with respect to the number of processed jobs (e.g. the number of cleaned items that models impurity degree of the pickling bath) and decreasing with respect to the additional resource allocated to a job, e.g. electrical energy that reduces the negative results of the aging effect (see [22]). We prove some properties of the analysed problems, and on their basis, we construct optimal polynomial time algorithms that can be easily used by practitioners. Since some of the proposed algorithms are dedicated for problems with arbitrary aging/deteriorating characteristics, then in such cases the complete knowledge about the optimized system (job parameters) is not required to construct an efficient schedule.

Although the considerations presented in this paper are dedicated to the optimization of pickling and galvanization processes, they can be applied for other similar problems that can be found in many other manufacturing systems, in which, for instance, tiredness of human workers (e.g. [6, 7]) or tool wear of lathe machines (e.g. [26]) affect the production output.

The remainder of this paper is organized as follows: Section 2 contains the problem formulation. Properties of the problems and resulting optimal polynomial time algorithms are provided in Section 3. The last section concludes the paper.

2 Problem formulation

There are given a set $J=\{1,\ldots,n\}$ of n jobs (e.g. pickling of individual metal items or batches of such metal

items that can be cleaned together in the same bath) and m machines (e.g. the pickling baths or cleaning stages), namely $M=\{M_1,\ldots, M_m\}$. Each job j consists of a set $O=\{O_{1,j},\ldots,O_{m,j}\}$ of m operations (e.g. cleaning activities related with particular stages). Each operation $O_{z,j}$ has to be processed on machine M_z ($z=1,\ldots,m$). Jobs (operations $O_{1,j}$) are available for processing on M_1 at time 0, and operation $O_{z+1,j}$ may start only if $O_{z,j}$ is completed. Due to the technological constraints (e.g. the order of cleaning activities such as polishing and pickling is fixed and the same for each cleaned metal item), it is assumed that machines have to process jobs in the same order; it is called a permutation flowshop. Since the capacity of a bath is limited and to avoid undesired chemical reactions between items of different metal kinds that occur if they are cleaned together, it is required that each machine can process at most one job (e.g. metal items of the same kind) at a time. Moreover, the technological requirements usually force that jobs cannot be preempted. We also assume that there are no precedence constraints between jobs. Further, instead of operation $O_{z,j}$, we say job j on machine M_z.

To model the aging effect, the processing time $p_j^{(z)}(v)$ of job j that is scheduled in position v in a sequence on machine M_z ($z=1,\ldots,m$) is defined as a non-decreasing positive function dependent on its position v (i.e. the number of previously machined products, $v-1$). Moreover, each job j is characterized by its normal processing time $a_j^{(z)}$ on machine M_z that is defined as the time required to process a job if the machine is not affected by aging (e.g. a fresh cleaning bath), i.e. $a_j^{(z)} \triangleq p_j^{(z)}(1)$.

In practice, it can be difficult to provide the exact shape of the aging characteristic $p_j^{(z)}(v)$; therefore, we consider aging models, where the processing time of job j on machine M_z is described by one of the following:

$$p_j^{(z)}(v) = a_j^{(z)} \cdot f(v), \tag{1}$$

$$p_j^{(z)}(v) = a_j^{(z)} + h(v), \tag{2}$$

$$p_j^{(z)}(v) = a_j^{(z)} + b_j^{(z)} v, \tag{3}$$

where $b_j^{(z)}$ is the linear aging ratio of job j on machine M_z, $f(v)$ and $h(v)$ are positive non-decreasing functions dependent on a job position in a sequence and $f(1)=1$ and $h(1)=0$ for $z=1,\ldots,m$ and $j=1,\ldots,n$. Observe that model 1 includes in particular model $p_j(v)=a_j v^{\alpha}$ (see [3]) as a special case (where α is an exponential aging ratio common for all jobs). These characteristics can approximate more complex aging functions; therefore, they can be applied by practitioners more likely.

In the further part of the paper, we use the following notation for the special cases of these functions: If $a_j^{(z)}=a_j^{(w)}$ and $b_j^{(z)}=b_j^{(w)}$ for $j=1,\ldots,n$ and $w,z=1,\ldots,m$ ($z\neq w$), then $a_j^{(z)}=a_j$ and $b_j^{(z)}=b_j$, respectively; if $a_j^{(z)}=a_k^{(z)}$ and $b_j^{(z)}=b_k^{(z)}$ for $j,k=1,\ldots,n$ ($j\neq k$) and $z=1,\ldots,m$, then $a_j^{(z)}=a^{(z)}$ and $b_j^{(z)}=b^{(z)}$, respectively; if $a_j^{(z)}=a_k^{(w)}$ and $b_j^{(z)}=b_k^{(w)}$ for $j,k=1,\ldots,n$ ($j\neq k$) and $z,w=1,\ldots,m$, then $a_j^{(z)}=a$ and $b_j^{(z)}=b$, respectively.

For the m-machine *permutation* flowshop problems, the schedule of jobs on the machines can be unambiguously defined by their sequence (permutation). Let $\pi=\langle\pi(1),\ldots,\pi(i),\ldots,\pi(n)\rangle$ denote the sequence of jobs on machines (permutation of the elements of the set J), where $\pi(i)$ is the job processed in position i in this sequence. By Π we will denote the set of all such permutations. Thus, for the given sequence $\pi\in\Pi$, we can determine the completion time $C_{\pi(i)}^{(z)}$ of a job scheduled in the ith position in a sequence π on machine M_z as follows:

$$C_{\pi(i)}^{(z)} = \max\left\{C_{\pi(i)}^{(z-1)}, C_{\pi(i-1)}^{(z)}\right\} + p_{\pi(i)}^{(z)}(i), \tag{4}$$

where $C_{\pi(1)}^{(0)} = C_{\pi(0)}^{(z)} = 0$ for $z=1,\ldots,m$ and $C_{\pi(i)}^{(1)} = \sum_{l=1}^{i} p_{\pi(l)}^{(1)}(l)$ is the completion time of a job placed in position i in the permutation π on M_1.

The objective is to find such a schedule π of jobs on the machines that minimizes the maximum completion time (makespan) $C_{\max}(\pi) \triangleq \max_{i=1,\ldots,n,z=1,\ldots,m}\{C_{\pi(i)}^{(z)}\}$ that is $C_{\max}(\pi) \triangleq C_{\pi(n)}^{(m)}$. Formally, the optimal schedule π^* for the makespan minimization problem is defined as follows: $\pi^* \triangleq \operatorname{argmin}_{\pi\in\Pi}\{C_{\pi(n)}^{(m)}\}$. The m-machine flowshop problems with models 1, 2 and 3 according to the three field notation scheme $X \mid Y \mid Z$ will be denoted as $Fm|p_j^{(z)}(v)=a_j^{(z)}\cdot f(v)|C_{\max}$, $Fm|p_j^{(z)}(v)=a_j^{(z)}+h(v)|C_{\max}$ and $Fm|p_j^{(z)}(v)=a_j^{(z)}+b_j^{(z)}v|C_{\max}$, respectively.

As was mentioned in the previous section, in many real-life problems, the negative influence of the aging effect on the time required to process a job can be decreased by the allocation of additional resources (e.g. fuel, electrical energy etc.). For instance, the wear of the pickling bath increases the cleaning time for metal items; however, it can be decreased for the items actually in the bath by heating it up. Let $u_j^{(z)}$ denote such an additional resource amount allocated to job j on machine M_z that can model electrical energy

or a gas, which heats up the pickling bath. On this basis and following [22], the processing time of job j on machine M_z, $p_j^{(z)}(v, u_j^{(z)})$, is given by a function dependent on position v of job j in a sequence and on the amount of an additional resource $u_j^{(z)}$ allocated to this job:

$$p_j^{(z)}\left(v, u_j^{(z)}\right) = p_j^{(z)}(v) - \gamma_j^{(z)} u_j^{(z)}, \tag{5}$$

where $p_j^{(z)}(v)$ is a non-decreasing function of job j dependent on its position v in a sequence and $\gamma_j^{(z)}$ is the resource ratio of job j on machine M_z that denotes the decreasing of its processing time depending on the amount of allocated resource $u_j^{(z)}$ to job j on machine M_z.

Apart from the presented above model with the aging effect $p_j^{(z)}(v)$ and resource allocation, we also analyse problems with the following model:

$$p_j^{(z)}\left(v, u_j^{(z)}\right) = a_j^{(z)} + b_j^{(z)} v - \gamma_j^{(z)} u_j^{(z)}. \tag{6}$$

Observe that the amount of a resource allocated to a job is usually limited in practice due to various technological restrictions, e.g. the range of the temperature of the pickling bath for each job is bounded. Therefore, it is assumed that there are constraints on the amount of a resource allocated to job j on machine M_z defined as follows: $\underline{u}_j^{(z)} \le u_j^{(z)} \le \overline{u}_j^{(z)}$, where $\underline{u}_j^{(z)}$ and $\overline{u}_j^{(z)}$ are the minimal and maximal amount of the resource that can be allocated to job j on machine M_z, respectively, and $0 \le \underline{u}_j^{(z)} \le \overline{u}_j^{(z)}$. Furthermore, the practical aspects (e.g. a company environmental policy or other financial requirements impose that there is a global restriction on the total resource consumption, $\hat{R}$, that can be modelled by $\sum_{j=1}^{n} \sum_{z=1}^{m} u_j^{(z)} \le \hat{R}$.

Note also that $\underline{u}_j^{(z)} > 0$ can be reduced to $\underline{u}_j^{(z)} = 0$ (for $j = 1, \ldots, n$ and $z = 1, \ldots, m$) by the following modification of parameters: $\hat{R} = \hat{R} - \sum_{j=1}^{n} \sum_{z=1}^{m} \underline{u}_j^{(z)}$, $\overline{u}_j^{(z)} = \overline{u}_j^{(z)} - \underline{u}_j^{(z)}$ and $p_j^{(z)}(v, u_j^{(z)}) = p_j^{(z)}(v, u_j^{(z)}) - \gamma_j^{(z)} \underline{u}_j^{(z)}$. Therefore, for convenience, we assume that $\underline{u}_j^{(z)} = 0$ for $j = 1, \ldots, n$ and $z = 1, \ldots, m$.

Now we will formulate the problems with the aging effect and resource allocation. Let $\mathbf{u} = \langle \mathbf{u^{(1)}}, \ldots, \mathbf{u^{(z)}}, \ldots, \mathbf{u^{(m)}} \rangle$ be the resource allocation, where $\mathbf{u^{(z)}} = \langle u_1^{(z)}, \ldots, u_j^{(z)}, \ldots, u_n^{(z)} \rangle$ is the resource allocation to jobs on machine M_z and recall that $u_j^{(z)}$ is the resource amount allocated to job j on machine M_z.

On this basis, for a given schedule π and resource allocation $\mathbf{u}$, we can determine the completion time of job $\pi(i)$ on machine M_z as follows:

$$C_{\pi(i)}^{(z)}(\mathbf{u}) = \max\left\{C_{\pi(i)}^{(z-1)}(\mathbf{u}), C_{\pi(i-1)}^{(z)}(\mathbf{u})\right\} + p_{\pi(i)}^{(z)}\left(i, u_{\pi(i)}^{(z)}\right) \tag{7}$$

where $C_{\pi(i)}^{(0)}(\mathbf{u}) = C_{\pi(0)}^{(z)}(\mathbf{u}) = 0$ for $z = 1, \ldots, m$ and $C_{\pi(i)}^{(1)}(\mathbf{u}) = \sum_{l=1}^{i} p_{\pi(l)}^{(1)}(l, u_{\pi(l)}^{(1)})$ is the completion time of a job placed in position i in the permutation π on M_1 with the given amount of allocated resource $\mathbf{u}$.

The objective is to find such a schedule π and resource allocation $\mathbf{u}$ that minimize the makespan $C_{\max}(\pi, \mathbf{u}) \triangleq \max_{\pi \in \Pi, \mathbf{u} \in \mathbf{U}} \{C_{\pi(n)}^{(m)}(\mathbf{u})\}$, where $\mathbf{U}$ denotes the set of all feasible resource allocations. Formally, the optimal schedule π^* and resource allocation $\mathbf{u}^*$ for the makespan minimization problem is defined as follows: $(\pi^*, \mathbf{u}^*) \triangleq \operatorname{argmin}_{\pi \in \Pi, \mathbf{u} \in \mathbf{U}} \{C_{\pi(n)}^{(m)}(\mathbf{u})\}$.

Above we have introduced a new general model of flowshop scheduling problems with the aging effect and additional resource allocation that can describe many real-life settings, where the negative influence of the aging effect on different production stages can be counteracted by the allocation of additional resource. However, in the further part, we will analyse its special cases with two machines, where job processing times on M_2 are constant, i.e. $p_j^{(2)}(v) = a_j^{(2)}$ and $\overline{u}_j^{(2)} = 0$ (thereby $u_j^{(2)} = 0$ and $\gamma_j^{(2)} = 0$) for $j, v = 1, \ldots, n$. Thus, there is no aging nor possibility to allocate additional resource for M_2, i.e. $p_j^{(2)}(v, u_j^{(2)}) = a_j^{(2)}$ (for $j, v = 1, \ldots, n$). However, it models following real-life settings, e.g. pickling (affected by aging and alternatively improved by the allocation of additional resource) is the first stage and galvanization (with constant job processing times) is the second stage of a manufacturing process.

Since the resource allocation is related only with M_1, therefore, for convenience, we will use the following notation: $u_j^{(1)} = u_j$, $\overline{u}_j^{(1)} = \overline{u}_j$, $\gamma_j^{(1)} = \gamma_j$ (for $j = 1, \ldots, n$), thereby $\mathbf{u} \triangleq \mathbf{u^{(1)}} = \langle u_1, \ldots, u_j, \ldots, u_n \rangle$ and $\sum_{j=1}^{n} u_j \le \hat{R}$; furthermore $b^{(2)} = 0$ for Eq. 6 and $m = 2$. The analysed problems in the three-field notation scheme will be denoted as $F2|p_j^{(1)} = a^{(1)} + b^{(1)} v - \gamma_j u_j, p_j^{(2)}(v, u_j) = a^{(2)}, \bar{u}_j = \overline{u}, \sum u_j \le \hat{R}|C_{\max}$, $F2|p_j^{(1)}(v, u_j) = a_j^{(1)} + b^{(1)} v - \gamma u_j, p_j^{(2)}(v, u_j) = a^{(2)}, \sum u_j \le \hat{R}|C_{\max}$ and $F2|p_j^{(1)}(v, u_j) = p^{(1)}(v) - \gamma u_j, p_j^{(2)}(v, u_j) = a_j^{(2)}, \bar{u}_j = \overline{u}, \sum u_j \le \hat{R}|C_{\max}$.

3 Properties and solution algorithms

In this section, we will provide optimal polynomial time algorithms that solve the problems formulated in

Section 2. First we will analyse problems with the aging effect only and next cases, where the aging effect can be decreased by the allocation of additional resource.

3.1 Problems without additional resource allocation

In this part, we will analyse problems with the aging effect only. In such cases, the system objectives cannot be improved by the additional resources; however, it can be done in a limited range by the schedule of processed jobs.

At first, we will show the Johnson rule [13] that is optimal for the classical two-machine flowshop problem with the makespan minimization is no longer optimal even for a simple linear aging model, i.e. $p_j^{(z)}(v) = a_j^{(z)} + b\,v$ for $z = 1, \ldots, m$ and $j = 1, \ldots, n$.

Example 1 Consider the following instance of the problem $F2|p_j^{(z)}(v) = a_j^{(z)} + b\,v|C_{\max}$: $a_1^{(1)} = 4$, $a_1^{(2)} = 8$, $a_2^{(1)} = 3$, $a_2^{(2)} = 3$, $a_3^{(1)} = 3$, $a_3^{(2)} = 4$, $a_4^{(1)} = 1$, $a_4^{(2)} = 4$, $a_5^{(1)} = 8$, $a_5^{(2)} = 7$ and $b = 2$. The optimal criterion value is $C_{\max}(\pi^*) = 63$ for $\pi^* = \{4, 1, 5, 3, 2\}$; on the other hand, Johnson's algorithm provides a schedule $\pi = \{4, 3, 1, 5, 2\}$ with $C_{\max}(\pi) = 64$.

Now, we will provide some useful expressions. Namely, following [1] and [14], the completion time of a job scheduled in the ith position in π on machine M_2 equals

$$C_{\pi(i)}^{(2)} = \sum_{l=1}^{i} p_{\pi(l)}^{(2)}(l) + \sum_{l=1}^{i} I_{\pi(l)}, \tag{8}$$

where $I_{\pi(l)}$ denotes the idle time on M_2 immediately before the start time of $\pi(l)$, i.e. between finishing job $\pi(l-1)$ and starting $\pi(l)$ for $l = 1, \ldots, i$. The total idle time on M_2 for the first i jobs can be calculated in the following way:

$$\sum_{l=1}^{i} I_{\pi(l)} = \max_{1 \le v \le i}\{Y_{\pi(v)}\},$$

where $Y_{\pi(v)}$ is equal to

$$Y_{\pi(v)} = \sum_{l=1}^{v} p_{\pi(l)}^{(1)}(l) - \sum_{l=1}^{v-1} p_{\pi(l)}^{(2)}(l),$$

thus, we have

$$\sum_{l=1}^{i} I_{\pi(l)} = \max_{1 \le v \le i}\left\{\sum_{l=1}^{v} p_{\pi(l)}^{(1)}(l) - \sum_{l=1}^{v-1} p_{\pi(l)}^{(2)}(l)\right\}. \tag{9}$$

Next, we will consider problems with ordered job processing times that application to model practical settings was justified in [25]. Namely, we will say that job processing times are ordered if $a_j^{(1)} > a_j^{(2)}$ for $j = 1, \ldots, n$ and $a_j^{(1)} > a_k^{(1)}$ implies $a_j^{(2)} > a_k^{(2)}$ for $j, k \in \{1, \ldots, n\}$ and $j \neq k$; in the three-field notation scheme, it will be denoted by ord.

On the basis of Eqs. 8 and 9 and using the well-known job interchanging technique, we can easily prove the following property:

Property 1 The problem $F2|p_j^{(z)}(v) = a_j^{(z)} \cdot f(v), ord|C_{\max}$ can be solved optimally in $O(n \log n)$ steps by scheduling jobs according to the non-increasing order of the normal job processing times $a_j^{(z)}$ for fixed z ($z \in \{1, 2\}$).

Proof The proof will be done in the same way as a proof for a similar problem with the learning effect by Koulamas and Kyparisis [14].

Assume there is given an optimal permutation π, which does not hold the rule from the thesis of this property. Therefore, for this permutation, there exists a pair of jobs $\pi(i)$ and $\pi(i+1)$, where $a_{\pi(i)}^{(z)} < a_{\pi(i+1)}^{(z)}$ for $z = 1, 2$. Assume there is given a permutation π', which has been obtained from the permutation π by interchanging the jobs from the ith and the $(i+1)$th position. It is easy to notice that $p_{\pi'(l)}^{(z)}(l) = p_{\pi(l)}^{(z)}(l)$ and $Y_{\pi'(l)} = Y_{\pi(l)}$ for $l = 1, \ldots i-1$ and $z = 1, 2$. Based on Eqs. 8 and 9, we have

$$\begin{aligned} C_{\pi(i+1)}^{(2)} - C_{\pi'(i+1)}^{(2)} &= \sum_{l=1}^{i+1} p_{\pi(l)}^{(2)}(l) + \max\{Y_{\pi(i)}, Y_{\pi(i+1)}\} \\ &\quad - \sum_{l=1}^{i+1} p_{\pi'(l)}^{(2)}(l) - \max\{Y_{\pi'(i)}, Y_{\pi'(i+1)}\} \\ &= \left(a_{\pi(i+1)}^{(2)} - a_{\pi(i)}^{(2)}\right)\left(f(i+1) - f(i)\right) \\ &\quad + \max\{Y_{\pi(i)}, Y_{\pi(i+1)}\} \\ &\quad - \max\{Y_{\pi'(i)}, Y_{\pi'(i+1)}\}. \end{aligned}$$

Since $a_{\pi(i)}^{(2)} < a_{\pi(i+1)}^{(2)}$ and $a_{\pi(i+1)}^{(1)} > a_{\pi(i)}^{(1)} > a_{\pi(i)}^{(2)}$ and $\max\{Y_{\pi(i)}, Y_{\pi(i+1)}\} = Y_{\pi(i+1)}$, then $C_{\pi(i+1)}^{(2)} - C_{\pi'(i+1)}^{(2)} \ge Y_{\pi(i+1)} - \max\{Y_{\pi'(i)}, Y_{\pi'(i+1)}\}$. Thus, only two cases have to be considered: (a) $Y_{\pi'(i)} > Y_{\pi'(i+1)}$ and (b) $Y_{\pi'(i)} < Y_{\pi'(i+1)}$ for which we have:

a) $$\begin{aligned} Y_{\pi(i+1)} - Y_{\pi'(i)} &= \left(a_{\pi(i)}^{(1)} - a_{\pi(i)}^{(2)}\right) f(i) \\ &\quad + a_{\pi(i+1)}^{(1)}\left(f(i+1) - f(i)\right) > 0, \end{aligned}$$

b) $Y_{\pi(i+1)} - Y_{\pi'(i+1)} = \left(a^{(1)}_{\pi(i+1)} - a^{(1)}_{\pi(i)}\right) \times (f(i+1) - f(i)) + \left(a^{(2)}_{\pi(i+1)} - a^{(2)}_{\pi(i)}\right) f(i) > 0.$

For both cases, $C^{(2)}_{\pi(i+1)} - C^{(2)}_{\pi'(i+1)} > 0$; thus, the permutation π cannot be optimal and the non-increasing order of $a^{(z)}_j$ (for fixed $z \in \{1, 2\}$) gives the optimal solution to the considered problem. □

The following properties can be proved in the similar manner:

Property 2 The problem $F2|p^{(z)}_j(v) = a^{(z)}_j + h(v), ord|C_{\max}$ can be solved optimally in $O(n \log n)$ steps by scheduling jobs according to the non-increasing order of the normal job processing times $a^{(z)}_j$ for fixed z ($z \in \{1, 2\}$).

Property 3 The problem $F2|p^{(z)}_j(v) = a + b^{(z)}_j v, b^{(2)}_j = 0|C_{\max}$ can be solved optimally in $O(n \log n)$ steps by scheduling jobs according to the non-increasing order of the ratios $b^{(1)}_j$.

Now, we will focus on the m-machine proportional flowshop problem, where the processing times of a job are identical on each machine, i.e. $p^{(z)}_j(v) = p_j(v)$ for $z = 1, \ldots, m$ and $j, v = 1, \ldots, n$. Proportional flowshop problems were justified by Pinedo in [21].

Property 4 The problem $Fm|p^{(z)}_j(v) = a_j \cdot f(v)|C_{\max}$ can be solved optimally in $O(n \log n)$ steps by scheduling jobs according to the non-increasing order of the normal job processing times a_j.

Proof The proof is similar to that provided by Wang and Xia [30] for a problem with the learning effect. Thus, based on [21] and Eq. 7, the makespan for $Fm|p^{(z)}_j(v) = a_j \cdot f(v)|C_{\max}$ can be given by:

$$C_{\max} = \sum_{i=1}^{n} a_{\pi(i)} f(i) + (m-1)\max\{a_{\pi(1)} f(1), \ldots, a_{\pi(i)} f(i), \ldots, a_{\pi(n)} f(n)\}. \quad (10)$$

Assume now that π is optimal and it does not hold the rule from the thesis of this property. Therefore, for this permutation, there exists a pair of jobs $\pi(i)$ and $\pi(i+1)$, where $a_{\pi(i)} < a_{\pi(i+1)}$. Assume there is given a permutation π', which has been obtained from the permutation π by interchanging the jobs from the ith and the $(i+1)$th position. Based on expression 10, we have

$$C^{(m)}_{\pi(n)} - C^{(m)}_{\pi'(n)} = \left(a_{\pi(i+1)} - a_{\pi(i)}\right)(f(i+1) - f(i)) + (m-1)\max \times \left\{a_{\pi(1)} f(1), \ldots, a_{\pi(i)} f(i), a_{\pi(i+1)} \times f(i+1), \ldots, a_{\pi(n)} f(n)\right\} - (m-1)\max \times \left\{a_{\pi(1)} f(1), \ldots, a_{\pi(i+1)} f(i), a_{\pi(i)} \times f(i+1), \ldots, a_{\pi(n)} f(n)\right\}.$$

Since $a_{\pi(i)} < a_{\pi(i+1)}$ and $a_{\pi(i+1)} f(i+1) > a_{\pi(i)} f(i+1)$ and $a_{\pi(i+1)}(i+1) > a_{\pi(i+1)}(i)$, then $C^{(m)}_{\pi(n)} - C^{(m)}_{\pi'(n)} > 0$. Thus, π cannot be optimal and the non-increasing order of the normal job processing times a_j gives the optimal solution to the considered problem. □

The following properties can be proved in the same way:

Property 5 The problem $Fm|p^{(z)}_j(v) = a_j + h(v)|C_{\max}$ can be solved optimally in $O(n \log n)$ steps by scheduling jobs according to the non-increasing order of the normal job processing times a_j.

Property 6 The problem $Fm|p^{(z)}_j(v) = a + b_j v|C_{\max}$ can be solved optimally in $O(n \log n)$ steps by scheduling jobs according to the non-increasing order of the ratios b_j.

3.2 Problems with additional resource allocation

In this section, we will analyse the problems with the aging effect and resource allocation. However, note that the two-machine permutation flowshop problem with additional resource allocation only under the makespan minimization is NP-hard if resource ratios are identical on M_1 and job processing times are constant on M_2 (see [11]), i.e. $F2|p^{(1)}_j(u_j) = a^{(1)}_j - \gamma u_j, p^{(2)}_j = a^{(2)}_j, \sum u_j \leq \hat{R}|C_{\max}$. Nevertheless, we will prove properties of optimal solutions for the cases formulated in Section 2. On this basis, we will provide for them optimal polynomial time algorithms.

First, we provide useful expressions that are based on Eqs. 8 and 9. Namely, the completion time of a job scheduled in position i for the resource allocation $\mathbf{u}$ can be rewritten as:

$$C^{(2)}_{\pi(i)}(\mathbf{u}) = \sum_{i=1}^{i} a^{(2)}_{\pi(l)} + \max_{1 \leq v \leq i}\{Y_{\pi(v)}(\mathbf{u})\}, \quad (11)$$

where

$$Y_{\pi(v)}(\mathbf{u}) = \sum_{l=1}^{v} p^{(1)}_{\pi(l)}(v) - \sum_{l=1}^{v} \gamma_{\pi(l)} u_{\pi(l)} - \sum_{l=1}^{v-1} a^{(2)}_{\pi(l)}. \quad (12)$$

On this basis, we prove the following properties:

Property 7 The problem $F2|p^{(1)}_j = a^{(1)} + b^{(1)}v - \gamma_j u_j, p^{(2)}_j(v, u_j) = a^{(2)}, \bar{u}_j = \bar{u}, \sum u_j \le \hat{R}|C_{\max}$ can be solved optimally in $O(n \log n)$ steps by allocating the resource in the non-increasing maximal possible amount starting from the jobs with the maximal value of parameter γ_j and by scheduling jobs according to the non-increasing values of resource ratios γ_j.

Proof To prove that the solution consistent with this property is optimal, at first we will show that the resource should be allocated up to the maximal possible amount starting from the jobs with the maximal value of parameter γ_j. Assume there is given a fixed arbitrary schedule π. Furthermore, assume there is given an optimal resource allocation $\mathbf{u}$, which does not comply with the thesis of this property, i.e. there exists at least one pair of jobs $\pi(h)$ and $\pi(k)$ $(1 \le h < k \le n)$ with $u_{\pi(h)} \le u_{\pi(k)}$ for which $\gamma_{\pi(h)} > \gamma_{\pi(k)}$. Let $\mathbf{u}'$ denote a resource allocation obtained from $\mathbf{u}$ by interchanging the amount of resource allocated to jobs $\pi(h)$ and $\pi(k)$. Since the amount of resource allocated to jobs scheduled in positions $1, \ldots, i-1$ is the same in $\mathbf{u}$ and $\mathbf{u}'$, thus the difference $C^{(2)}_{\pi(k)}(\mathbf{u}) - C^{(2)}_{\pi(k)}(\mathbf{u}')$ can be calculated as follows:

$$C^{(2)}_{\pi(k)}(\mathbf{u}) - C^{(2)}_{\pi(k)}(\mathbf{u}') = \max_{h \le v \le k}\{Y_{\pi(v)}(\mathbf{u})\} - \max_{h \le v \le k}\{Y_{\pi(v)}(\mathbf{u}')\}.$$

The amount of resource allocated to jobs scheduled in positions $h+1, \ldots, k-1$ is the same in $\mathbf{u}$ and $\mathbf{u}'$; thus, it is sufficient to compare the following two differences:

$$Y_{\pi(h)}(\mathbf{u}) - Y_{\pi(h)}(\mathbf{u}') = \gamma_{\pi(h)}(u_{\pi(k)} - u_{\pi(h)}) \ge 0 \text{ and}$$

$$Y_{\pi(k)}(\mathbf{u}) - Y_{\pi(k)}(\mathbf{u}') = (\gamma_{\pi(h)} - \gamma_{\pi(k)})(u_{\pi(k)} - u_{\pi(h)}) \ge 0.$$

Since both of them are non-negative, then we can conclude that:

$$\max_{h \le v \le k}\{Y_{\pi(v)}(\mathbf{u})\} - \max_{h \le v \le k}\{Y_{\pi(v)}(\mathbf{u}')\} \ge 0$$

and $C^{(2)}_{\pi(k)}(\mathbf{u}) - C^{(2)}_{\pi(k)}(\mathbf{u}') \ge 0$; hence, the criterion value of $\mathbf{u}'$ cannot be greater than that of $\mathbf{u}$. Therefore, in the optimal solution, the resource is allocated in the non-increasing amount starting from the jobs with the maximal value of γ_j.

Now, we have to prove that the resource should be allocated to the maximal possible extent. Assume there are given a permutation π and an optimal resource allocation $\mathbf{u}'$, where the resource is allocated according to non-increasing values of γ_j, but not up to the maximal possible value. Thus, there exists at least two jobs $\pi(h)$ and $\pi(k)$ $(1 \le h < k \le n)$ for which $u'_{\pi(h)} = \bar{u} - \Delta_{\pi(h)}$, $u'_{\pi(k)} = \bar{u} - \Delta_{\pi(k)}$, $\gamma_{\pi(h)} > \gamma_{\pi(k)}$ and $\bar{u} > \Delta_{\pi(k)} > \Delta_{\pi(h)} > 0$ and $\Delta_{\pi(h)} + \Delta_{\pi(k)} \le \bar{u}$ follows from $u'_{\pi(h)} + u'_{\pi(k)} \ge \bar{u}$; otherwise, there is not enough resource to allocate it up to the maximal possible amount. Note that if $\Delta_{\pi(h)} + \Delta_{\pi(k)} > \bar{u}$, i.e. $u'_{\pi(h)} + u'_{\pi(k)} < \bar{u}$, then we can assume that the maximum possible amount is equal to the amount of the available resource, i.e. $\bar{u} = u'_{\pi(h)} + u'_{\pi(k)}$ and $\Delta_{\pi(h)}$, $\Delta_{\pi(k)}$ are calculated for the new $\bar{u}$.

Let $\mathbf{u}''$ denote a resource allocation obtained from $\mathbf{u}$, such that the only difference between them is the amount of resource allocated to jobs in positions h and k $(1 \le h < k \le n)$ that is equal to $u''_{\pi(h)} = \bar{u}$ and $u''_{\pi(k)} = \bar{u} - \Delta_{\pi(h)} - \Delta_{\pi(k)}$. As in the previous case, since the amount of resource allocated to jobs in positions $1, \ldots, h-1$ is the same, we have:

$$C^{(2)}_{\pi(k)}(\mathbf{u}') - C^{(2)}_{\pi(k)}(\mathbf{u}'') = \max_{h \le v \le k}\{Y_{\pi(v)}(\mathbf{u}')\} - \max_{h \le v \le k}\{Y_{\pi(v)}(\mathbf{u}'')\}.$$

The amount of resource allocated to jobs scheduled in positions $h+1, \ldots, k-1$ is the same in $\mathbf{u}'$ and $\mathbf{u}''$; thus, it is sufficient to compare the following two differences:

$$Y_{\pi(h)}(\mathbf{u}') - Y_{\pi(h)}(\mathbf{u}'') = \gamma_{\pi(h)}\Delta_{\pi(h)} > 0 \text{ and}$$

$$Y_{\pi(k)}(\mathbf{u}') - Y_{\pi(k)}(\mathbf{u}'') = \Delta_{\pi(h)}(\gamma_{\pi(h)} - \gamma_{\pi(k)}) > 0.$$

Both of them are positive; thus, $C^{(2)}_{\pi(k)}(\mathbf{u}') - C^{(2)}_{\pi(k)}(\mathbf{u}'') > 0$; hence, the criterion value of $\mathbf{u}''$ cannot be greater than that of $\mathbf{u}'$. Therefore, allocating the resource according to non-increasing values of parameter γ_j up to the maximal possible amount is optimal.

Now, we will prove that in the optimal solution, jobs should be scheduled according to the non-increasing values of parameter γ_j. Assume there is given an optimal fixed resource allocation $\mathbf{u}''$ and a permutation π, which does not comply with the thesis of this property, i.e. there exists at least two jobs $\pi(i)$ and $\pi(i+1)$ for which $\gamma_{\pi(i)} < \gamma_{\pi(i+1)}$. Assume there is given a permutation π', which has been obtained from π by interchanging jobs from the ith and $(i+1)$th positions. Since the sequence of jobs in positions $1, \ldots, i-1$ is the same in π and π', thus we have:

$$C^{(2)}_{\pi(i+1)}(\mathbf{u}'') - C^{(2)}_{\pi'(i+1)}(\mathbf{u}'') = \max\{Y_{\pi(i)}(\mathbf{u}''), Y_{\pi(i+1)}(\mathbf{u}'')\} - \max\{Y_{\pi'(i)}(\mathbf{u}''), Y_{\pi'(i+1)}(\mathbf{u}'')\}.$$

On this basis and knowing that

$$Y_{\pi(i)}(\mathbf{u}'') - Y_{\pi'(i)}(\mathbf{u}'') = \gamma_{\pi(i+1)}u''_{\pi(i+1)} - \gamma_{\pi(i)}u''_{\pi(i)} > 0$$

and $Y_{\pi(i+1)}(\mathbf{u}'') = Y_{\pi'(i+1)}(\mathbf{u}'')$, the difference $C^{(2)}_{\pi(i+1)}(\mathbf{u}'') - C^{(2)}_{\pi(i+1)}(\mathbf{u}'')$ is non-negative; hence, the criterion value of π' cannot be greater than that of π. Therefore, scheduling jobs according to non-increasing values of γ_j provides the optimal sequence for the given optimal resource allocation. □

Algorithm 1

```
1: Determine initial solution π_init by
   scheduling jobs in the non-decreasing
   order of p_j^(1)(1, min{ū_j, Û}), π*=∅, u*=0
2: For i=1 To n
       Assign π*(i)=π_init(i), u*_π*(i)=min{ū*_π*∅,Û},
       Û = Û − u*_π*(i)
3: π* is an optimal schedule and u* is
   an optimal resource allocation
```

Property 8 The problem $F2|p^{(1)}_j(v,u_j) = a^{(1)}_j + b^{(1)}v - \gamma u_j, p^{(2)}_j(v,u_j) = a^{(2)}, \sum u_j \le \hat{R}|C_{\max}$ can be solved optimally in $O(n\log n)$ steps according to Algorithm 1.

Proof To prove this property, at first we will show that for the fixed arbitrary schedule, the resource should be allocated up to the maximal possible amount to the succeeding jobs starting from the first scheduled job. Next, we will prove that scheduling jobs according to the non-decreasing values of $p^{(1)}_j(1,\min\{\bar{u}_j,\hat{U}\})$ is optimal.

To prove that the resource is allocated to the maximal possible extent, we assume that for a fixed arbitrary permutation π the resource is allocated to succeeding jobs starting from the first scheduled job, but not up to the maximal possible amount. Assume that there is given an optimal resource allocation denoted by $\mathbf{u}$, where for at least two jobs $\pi(i)$ and $\pi(i+1)$, the resource is allocated not up to the maximal possible amount. Therefore, the resource allocation to jobs $\pi(i)$ and $\pi(i+1)$ is $u_{\pi(i)} = \bar{u}_{\pi(i)} - \Delta_{\pi(i)}$, $u_{\pi(i+1)} = \bar{u}_{\pi(i+1)} - \Delta_{\pi(i+1)}$, $\bar{u}_{\pi(i)} > \Delta_{\pi(i)} > 0$, $\bar{u}_{\pi(i+1)} > \Delta_{\pi(i+1)} > 0$ and $\Delta_{\pi(i)} + \Delta_{\pi(i+1)} \le \bar{u}_{\pi(i)}$, $\Delta_{\pi(i)} + \Delta_{\pi(i+1)} \le \bar{u}_{\pi(i+1)}$ follows from $u_{\pi(i)} + u_{\pi(i+1)} \ge \bar{u}_{\pi(i)}$ and $u_{\pi(i)} + u_{\pi(i+1)} \ge \bar{u}_{\pi(i+1)}$, respectively; otherwise, there is not enough resource to allocate it up to the maximal possible amount. Note that if $\Delta_{\pi(i)} + \Delta_{\pi(i+1)} > \bar{u}_{\pi(i)}$ and $\Delta_{\pi(i)} + \Delta_{\pi(i+1)} > \bar{u}_{\pi(i+1)}$, i.e. $u_{\pi(i)} + u_{\pi(i+1)} < \bar{u}_{\pi(i)}$ and $u_{\pi(i)} + u_{\pi(i+1)} < \bar{u}_{\pi(i+1)}$, then we can assume that the maximum possible amount is equal to the amount of the available resource, i.e. $\bar{u}_{\pi(i)} = \bar{u}_{\pi(i+1)} = u_{\pi(i)} + u_{\pi(i+1)}$ and $\Delta_{\pi(i)}$, $\Delta_{\pi(i+1)}$ are calculated for the new $\bar{u}_{\pi(i)}$ and $\bar{u}_{\pi(i+1)}$, respectively.

Let us construct a new resource allocation $\mathbf{u}'$, which is obtained from $\mathbf{u}$, such that the only difference between them is the resource allocation to jobs $\pi(i)$ and $\pi(i+1)$, which is equal to $u'_{\pi(i)} = \bar{u}_{\pi(i)}$ and $u'_{\pi(i+1)} = \bar{u}_{\pi(i+1)} - \Delta_{\pi(i)} - \Delta_{\pi(i+1)}$.

The amount of resource allocated to jobs in positions $1,\ldots,i-1$ is the same; thus, we have:

$$C^{(2)}_{\pi(i+1)}(\mathbf{u}) - C^{(2)}_{\pi(i+1)}(\mathbf{u}') = \max\{Y_{\pi(i)}(\mathbf{u}), Y_{\pi(i+1)}(\mathbf{u})\} - \max\{Y_{\pi(i)}(\mathbf{u}'), Y_{\pi(i+1)}(\mathbf{u}')\},$$

$$Y_{\pi(i)}(\mathbf{u}) - Y_{\pi(i)}(\mathbf{u}') = \gamma\Delta_{\pi(i)} \ge 0, \text{ and}$$

$$Y_{\pi(i+1)}(\mathbf{u}) - Y_{\pi(i+1)}(\mathbf{u}') = 0.$$

Since the differences are non-negative, then $C^{(2)}_{\pi(k)}(\mathbf{u}) - C^{(2)}_{\pi(k)}(\mathbf{u}') \ge 0$; hence, the criterion value of $\mathbf{u}'$ cannot be greater than that of $\mathbf{u}$. Therefore, in the optimal solution, resource is allocated to the maximal possible amount to the succeeding jobs starting from the first scheduled job.

There is given an optimal resource allocation $\mathbf{u}'$. Assume that there is given an optimal permutation π which does not hold the rule from the thesis of this property, i.e. there exist at least two jobs i and $i+1$ for which $p_{\pi(i)}(1,\min\{\bar{u}_{\pi(i)},\hat{U}\}) < p_{\pi(i+1)}(1,\min\{\bar{u}_{\pi(i+1)},\hat{U}\})$. Based on π, we obtain a sequence π' in which jobs in positions i and $i+1$ are interchanged. Since the each job processed on M_2 has the same job processing time and $p^{(z)}_{\pi'(l)}(l,u'_{\pi'(l)}) = p^{(z)}_{\pi(l)}(l,u'_{\pi(l)})$ and $Y_{\pi'(l)}(\mathbf{u}') = Y_{\pi(l)}(\mathbf{u}')$ for $l = 1,\ldots i-1$ and $z = 1,2$, then the completion time of job scheduled in position $i+1$th is equal to:

$$C^{(2)}_{\pi(i+1)}(\mathbf{u}') - C^{(2)}_{\pi'(i+1)}(\mathbf{u}') = \max\{Y_{\pi(i)}(\mathbf{u}'), Y_{\pi(i+1)}(\mathbf{u}')\} - \max\{Y_{\pi'(i)}(\mathbf{u}'), Y_{\pi'(i+1)}(\mathbf{u}')\}.$$

Moreover, since the aging ratio $b^{(1)}_j$ is the same for each job, i.e. $b^{(1)}_j = b^{(1)}$, then $Y_{\pi(i+1)}(\mathbf{u}') = Y_{\pi'(i+1)}(\mathbf{u}')$. Therefore, it is sufficient to compare $Y_{\pi(i)}(\mathbf{u}')$ and $Y_{\pi'(i)}(\mathbf{u}')$:

$$\begin{aligned} Y_{\pi(i)}(\mathbf{u}') - Y_{\pi'(i)}(\mathbf{u}') &= \left(a_{\pi(i)} + b^{(1)}i - \gamma u'_{\pi(i)}\right) \\ &\quad - \left(a_{\pi(i+1)} + b^{(1)}i - \gamma u'_{\pi(i+1)}\right) \\ &= p^{(1)}_{\pi(i)}\left(i, u'^{(1)}_{\pi(i)}\right) - p^{(1)}_{\pi'(i)}\left(i, u'^{(1)}_{\pi(i)}\right). \end{aligned}$$

Due to the assumption $p^{(1)}_{\pi(i)}(1, \min\{\bar{u}_{\pi(i)}, \hat{U}\}) < p^{(1)}_{\pi(i+1)}(1, \min\{\bar{u}_{\pi(i+1)}, \hat{U}\})$, the difference $C^{(2)}_{\pi(i+1)}(\mathbf{u}') - C^{(2)}_{\pi'(i+1)}(\mathbf{u}')$ is non-negative; therefore, the criterion value of permutation π' cannot be greater than that of π. Thus, scheduling jobs according Algorithm 1 is optimal. □

Note that Janiak [11] proposed an algorithm with complexity $O(n^2)$ for the problem $F2|p^{(1)}_j(v,u_j) = a^{(1)}_j - \gamma u_j, p^{(2)}_j(v,u_j) = a^{(2)}, \sum u_j \le \hat{R}|C_{\max}$ (i.e. without the aging effect) that is a special case of the considered problem (Property 8). However, we have improved it (Algorithm 1) to solve also cases with the aging effect, and furthermore, we have decreased its complexity to $O(n \log n)$.

Property 9 The problem $F2|p^{(1)}_j(v,u_j) = p^{(1)}(v) - \gamma u_j, p^{(2)}_j(v,u_j) = a^{(2)}_j, \bar{u}_j = \bar{u}^{(1)}, \sum u_j \le \hat{R}|C_{\max}$, where $p^{(1)}(v)$ is an arbitrary non-decreasing function the same for all jobs, can be solved optimally in $O(n \log n)$ steps by scheduling jobs according to the non-increasing values of $a^{(2)}_j$ and then by allocating the resource to succeeding jobs up to the maximal possible amount starting from the first scheduled job.

Proof To prove this property, at first we will show that the optimal resource allocation is schedule independent. Then, it will be proved that for the fixed arbitrary resource allocation in the optimal solution, jobs are scheduled according to the non-increasing values of $a^{(2)}_j$. Next, we will prove that the resource should be allocated up to the maximal possible amount starting from the first scheduled job. According to Eqs. 11 and 12, the completion time of job scheduled in position i can be rewritten as follows:

$$C^{(2)}_{\pi(i)}(\mathbf{u}) = \sum_{l=1}^{i} a^{(2)}_{\pi(l)} + \max_{1\le v\le i}\left\{\sum_{l=1}^{v} p^{(1)}(l) - \gamma\sum_{l=1}^{v} u_{\pi(l)} - \sum_{l=1}^{v-1} a^{(2)}_{\pi(l)}\right\}$$

Observe that the resource is allocated regardless of the permutation (is job independent), since the criterion value is minimized if the expression

$$\max_{1\le v\le i}\left\{\sum_{l=1}^{v} p^{(1)}(l) - \gamma\sum_{l=1}^{v} u_{\pi(l)} - \sum_{l=1}^{v-1} a^{(2)}_{\pi(l)}\right\}$$

is maximized, i.e. the value of allocated resource is maximized.

On this basis, we can conclude that the considered problem can be solved optimally in the following way. At first, jobs are scheduled according to non-increasing values of $a^{(2)}_j$, and then resource is allocated to the maximal possible amount to succeeding jobs starting from the first scheduled job.

Assume that for the given resource allocation $\mathbf{u}$ there is given an optimal permutation π which does not comply with the thesis of this property, i.e. the following inequality holds: $a^{(2)}_{\pi(i-1)} < a^{(2)}_{\pi(i)}$. Assume there is given permutation π' that has been obtained from the permutation π by interchanging jobs in positions $i-1$ and i. Based on expression 11, we have:

$$\begin{aligned} &C^{(2)}_{\pi(i+1)}(\mathbf{u}) - C^{(2)}_{\pi'(i+1)}(\mathbf{u}) \\ &= \max_{1\le h\le i+1}\left\{\sum_{l=1}^{i+1} p^{(1)}(l) - \gamma\sum_{l=1}^{i+1} u_{\pi(l)} - \sum_{l=1}^{i-1} a^{(2)}_{\pi(l)} - a^{(2)}_{\pi(i)}\right\} \\ &\quad - \max_{1\le h\le i+1}\left\{\sum_{l=1}^{i+1} p^{(1)}(l) - \gamma\sum_{l=1}^{i+1} u_{\pi(l)} - \sum_{l=1}^{i-1} a^{(2)}_{\pi(l)} - a^{(2)}_{\pi(i+1)}\right\}. \end{aligned}$$

Since $a^{(2)}_{\pi(i)} < a^{(2)}_{\pi(i+1)}$, then the difference $C^{(2)}_{\pi(i+1)} - C^{(2)}_{\pi'(i+1)}$ is positive. Therefore, the criterion value of permutation π' is not greater than that of π and scheduling jobs according to the non-increasing values of $a^{(2)}_j$ gives the optimal schedule.

Now, we will prove that the resource is allocated up to the maximal possible amount starting from the first scheduled job. Assume that there are given a permutation π and the optimal allocation of the resource $\mathbf{u}$, which is not allocated in the non-increasing amount. Therefore, there exists at least one pair of jobs $\pi(i)$ and $\pi(i+1)$, for which $u_{\pi(i)} < u_{\pi(i+1)}$. Assume also that for the permutation π there is given a resource allocation $\mathbf{u}'$, which is obtained from $\mathbf{u}$ by interchanging the amount of resource allocated to jobs $\pi(i)$ and $\pi(i+1)$. Since the amount of resource allocated to jobs scheduled in positions $1, 2, \ldots, i-1$ is the same in $\mathbf{u}$ and $\mathbf{u}'$, then on the basis of expression 11:

$$\begin{aligned} C^{(2)}_{\pi(i+1)}(\mathbf{u}) - C^{(2)}_{\pi(i+1)}(\mathbf{u}') &= \max\{Y_{\pi(i)}(\mathbf{u}), Y_{\pi(i+1)}(\mathbf{u})\} \\ &\quad - \max\{Y_{\pi(i)}(\mathbf{u}'), Y_{\pi(i+1)}(\mathbf{u}')\}. \end{aligned}$$

It can be seen that $Y_{\pi(i+1)}(\mathbf{u}) = Y_{\pi(i+1)}(\mathbf{u}')$. Therefore, it is enough to compare $Y_{\pi(i)}(\mathbf{u}) - Y_{\pi(i)}(\mathbf{u}')$, i.e.

$$Y_{\pi(i)}(\mathbf{u}) - Y_{\pi(i)}(\mathbf{u}') = \gamma\left(u_{\pi(i+1)} - u_{\pi(i)}\right).$$

Since $u_{\pi(i)} < u_{\pi(i+1)}$, then the difference $C^{(2)}_{\pi(i+1)}(\mathbf{u}) - C^{(2)}_{\pi(i+1)}(\mathbf{u}')$ is non-negative. Therefore, the criterion value of $\mathbf{u}'$ cannot be greater than that of $\mathbf{u}$.

To prove that the resource is allocated to the maximal possible extent, we assume that for a fixed arbitrary permutation π the resource is allocated to succeeding jobs starting from the first scheduled job, but not up to the maximal possible amount. Assume that there is given an optimal resource allocation $\mathbf{u}'$, where for at least two jobs $\pi(i)$ and $\pi(i+1)$, the resource is allocated not up to the maximal possible amount; therefore, the resource allocation to these jobs is $u'_{\pi(i)} = \bar{u} - \Delta_{\pi(i)}$, $u'_{\pi(i+1)} = \bar{u} - \Delta_{\pi(i+1)}$, $\gamma_{\pi(i)} > \gamma_{\pi(i+1)}$ and $\bar{u} > \Delta_{\pi(i+1)} > \Delta_{\pi(i)} > 0$ and $\Delta_{\pi(i)} + \Delta_{\pi(i+1)} \leq \bar{u}$ follows from $u'_{\pi(i)} + u'_{\pi(i+1)} \geq \bar{u}$; otherwise, there is not enough resource to allocate it up to the maximal possible amount. Note that if $\Delta_{\pi(i)} + \Delta_{\pi(i+1)} > \bar{u}$, i.e. $u'_{\pi(i)} + u'_{\pi(i+1)} < \bar{u}$, then we can assume that the maximum possible amount is equal to the amount of the available resource, i.e. $\bar{u} = u'_{\pi(i)} + uv_{\pi(i+1)}$ and $\Delta_{\pi(i)}, \Delta_{\pi(i+1)}$ are calculated for the new $\bar{u}$. Let $\mathbf{u}''$ denote a resource allocation obtained from $\mathbf{u}'$, such that the only difference between them is the amount of resource allocated to jobs in positions i and $i+1$ that is equal to $u''_{\pi(i)} = \bar{u}$ and $u''_{\pi(i+1)} = \bar{u} - \Delta_{\pi(i)} - \Delta_{\pi(i+1)}$. Since the amount of resource allocated to jobs in positions $1, \ldots, i-1$ is the same, then we have:

$$C^{(2)}_{\pi(i+1)}(\mathbf{u}') - C^{(2)}_{\pi(i+1)}(\mathbf{u}'') = \max\{Y_{\pi(i)}(\mathbf{u}'), Y_{\pi(i+1)}(\mathbf{u}')\} - \max\{Y_{\pi(i)}(\mathbf{u}''), Y_{\pi(i+1)}(\mathbf{u}'')\}$$

and $Y_{\pi(i)}(\mathbf{u}') - Y_{\pi(i)}(\mathbf{u}'') = \gamma\Delta_{\pi(i)} > 0$ and

$$Y_{\pi(i+1)}(\mathbf{u}') - Y_{\pi(i+1)}(\mathbf{u}'') = 0.$$

Since the difference $C^{(2)}_{\pi(i+1)}(\mathbf{u}') - C^{(2)}_{\pi(i+1)}(\mathbf{u}'')$ is non-negative, then the criterion value of $\mathbf{u}''$ cannot be greater than that of $\mathbf{u}'$. Therefore, the allocation of resource in the non-increasing and maximal amount to the succeeding jobs starting from the first scheduled job is optimal. □

4 Conclusions

In this paper, we analysed real-life problems, which in particular occur in industrial processes that accompany metal finishing. To solve them, we developed a new model of flowshop scheduling problems that takes into consideration the processing times of jobs can increase with the number of processed jobs and decrease by the allocation of additional resource. We proved that some cases of the considered problems are still polynomially solvable and provided for them optimal solution algorithms that can be easily used by practitioners. Since some of the proposed algorithms are dedicated for problems with arbitrary aging/deteriorating characteristics, then in such cases the complete knowledge about the optimized system (job parameters) is not required to construct an efficient schedule. Furthermore, they can be also used as heuristics for more complex settings.

Acknowledgements We are grateful to the Editor and the Referees for their valuable comments on an earlier version of our paper. This work was supported by Polish Ministry of Science and Higher Education under grant no. N N519 408337.

References

1. Baker KR, Trietsch D (2009) Principles of sequencing an scheduling. Wiley, Hoboken
2. Behnamian J, Fatemi Ghomi S (2011) Hybrid flowshop scheduling with machine and resource-dependent processing times. Appl Math Model 35:1107–1123
3. Chang PC, Chen SH, Mani V (2009) A note on due-date assignment and single machine scheduling with a learning/aging effect. Int J Prod Econ 117:142–149
4. Cheng TCE, Ding Q, Lin BMT (2004) A concise survey of scheduling with time-dependent processing times. Eur J Oper Res 152:1–13
5. Chung YH, Tong LI (2011) Makespan minimization for m-machine permutation flowshop scheduling problem with learning considerations. Int J Adv Manuf Technol 56:355–367
6. Dababneh AJ, Swanson N, Shell RL (2001) Impact of added rest breaks on the productivity and well being of workers. Ergonomics 44:164–174
7. Eilon S (1964) On a mechanistic approach to fatigue and rest periods. Int J Prod Res 3:327–332
8. Groshart EC (2000) Pickling and acid dipping. Met Finish 98:183–193
9. Groshart EC (2000) Preparation of basis metals for plating. Met Finish 98:194–205
10. Gupta JND, Stafford JEF (2006) Flowshop scheduling research after five decades. Eur J Oper Res 169:699–711
11. Janiak A (1998) Minimization of the makespan in a two-machine problem under given resource constraints. Eur J Oper Res 107:325–337
12. Janiak A, Rudek R (2010) Scheduling jobs under an aging effect. J Oper Res Soc 61:1041–1048
13. Johnson SM (1954) Optimal two-and-three-stage production schedules. Nav Res Log 1:61–68
14. Koulamas C, Kyparisis GJ (2007) Single-machine and two-machine flowshop scheduling with general learning functions. Eur J Oper Res 178:402–407
15. Kuo WH, Yang DL (2008) Minimizing the makespan in a single-machine scheduling problem with the cyclic process of an aging effect. J Oper Res Soc 59:416–420

16. Lai PJ, Lee WC, Chen HH (2011) Scheduling with deteriorating jobs and past-sequence-dependent setup times. Int J Adv Manuf Technol 54:737–741
17. Lee WC, Wu CC, Liu HC (2009) A note on single-machine makespan problem with general deteriorating function. Int J Adv Manuf Technol 40:1053–1056
18. Liu P, Zhou X, Tang L (2010) Two-agent single-machine scheduling with position-dependent processing times. Int J Adv Manuf Technol 48:325–331
19. Mandich NV (2003) Overview of surface preparation of metals prior to finishing: part 2. Met Finish 101:33–58
20. Parthasarathy S, Rajendran C (1997) An experimental evaluation of heuristics for scheduling in a real-life flowshop with sequence-dependent setup times of jobs. Int J Prod Econ 49:255–263
21. Pinedo M (1995) Scheduling: theory, algorithms and systems. Prentice-Hall, Englewood Cliffs
22. Rudek A, Rudek R (2011) A note on optimization in deteriorating systems using scheduling problems with the aging effect and resource allocation models. Comput Math Appl 62:1870–1878
23. Rudek R (2011) Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect. Comput Ind Eng 61:20–31
24. Sheir L, Jarman R, Burstein G (1994) Corrosion. Butterworth-Heinemann, Oxford
25. Smith ML, Panwalkar SS, Dudek RA (1975) Flowshop sequencing problem with ordered processing time matrices. Manage Sci 21:544–549
26. Stanford M, Lister P (2004) Investigation into the relationship between tool-wear and cutting environments when turning EN32 steel. Ind Lubr Tribol 56:114–121
27. Wang JB, Liu LL (2009) Two-machine flow shop problem with effects of deterioration and learning. Comput Ind Eng 57:1114–1121
28. Wang JB, Wang LY, Wang D, Huang X, Wang XR (2009) A note on single-machine total completion time problem with general deteriorating function. Int J Adv Manuf Technol 44:1213–1218
29. Wang JB, Wang LY, Wang D, Wang XY (2009) Single-machine scheduling with a time-dependent deterioration. Int J Adv Manuf Technol 43:805–809
30. Wang JB, Xia ZQ (2005) Flow-shop scheduling with a learning effect. J Oper Res Soc 56:1325–1330
31. Xu Z, Sun L, Gong J (2008) Worst-case analysis for flow shop scheduling with a learning effect. Int J Prod Econ 113:748–753
32. Yang SJ, Yang DL (2010) Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities. OMEGA 38:528–533
33. Yang SJ, Yang DL (2010) Minimizing the total completion time in single-machine scheduling with aging/deteriorating effects and deteriorating maintenance activities. Comput Math Appl 60:2161–2169